VĚDECKÉ SPISY VYSOKÉHO UČENÍ TECHNICKÉHO V BRNĚ

Edice PhD Thesis, sv. 511
ISSN 1213-4198

Ing. Vítězslav Křivánek

Systémy realizace protichybového kódování

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

Ing. Vítězslav Křivánek

SYSTÉMY REALIZACE PROTICHYBOVÉHO KÓDOVÁNÍ

SYSTEMS DESIGN OF CORRECTION CODING

ZKRÁCENÁ VERZE PH.D. THESIS

Obor:	Teleinformatika
Školitel:	doc. Ing. Karel Němec, CSc.
Oponenti:	Prof. Ing. Florian Makáň, Ph.D. doc. Ing. Izabela Krbilová, CSc.
Datum obhajoby:	19. 1. 2009

KLÍČOVÁ SLOVA
Protichybové kódování, shlukové chyby, výběrová kritéria, konvoluční kódy, individuální protichybové systémy, počítačová simulace, Matlab, VHDL.

KEYWORDS
Forward Error Correction (FEC), Burst-Error, Selection Criterion, Convolution Coding, Individual Anti-Error Systems, Computer Simulation, Matlab, VHDL.

Disertační práce je k dispozici na Vědeckém oddělení děkanátu FEKT VUT v Brně, Údolní 53, Brno, 602 00

ISBN 978-80-214-3819-4
ISSN 1213-4198

OBSAH

1 ÚVOD

Zvláště v posledních letech roste potřeba efektivních a spolehlivých systémů přenosu dat. Hlavní příčinou je velmi rychlý rozmach vysokorychlostních systémů pro výměnu, zpracování a uchování dat. S tím souvisí i stálá expanze multimédií a internetu, jež přináší stále větší potřebu i požadavky na používané technologie a jejich vývoj. Pro návrh těchto systémů je zapotřebí slučování komunikačních i počítačových technologií. Jeden z hlavních zájmů spočívá v kontrole chyb a v následném obnovení získaných dat. Jelikož pro příjemce zprávy má význam pouze bezchybná zpráva, je zřejmá snaha přenášenou zprávu proti chybám zabezpečit. S rozvojem digitálních sítí nabývá téma zabezpečování přenášené informace na významu. Dnes se již standardně a zcela systematicky používají při přenosu zpráv v digitálních sítích bezpečnostní kódy.

Fyzikální prostředí, ve kterém je informace přenášena, se nazývá kanál. Při přenosu informace kanálem může dojít k jejímu poškození kvůli poruchám fyzikálního prostředí, ve kterém k přenosu dochází. Obecně se nazývají šum, jenž způsobuje, že přijatá zpráva se liší od zprávy vyslané. Všude tam, kde je nutné eliminovat vliv šumu, který je způsoben okolním prostředím, se uplatňují zabezpečovací kódy. V současnosti, jistě i do budoucna, je snaha u moderních komunikačních technologií neustále zrychlovat přenos dat. To s sebou přináší nové problémy, jež při takovém přenosu vznikají, a je třeba jejich odstranění. Především se při přenosu dat lze díky zvyšování přenosové rychlosti stále častěji setkat s chybami, které mají tendenci se shlukovat. Jedním z hlavních problémů je i úkol najít adekvátní kódování a opravu chyb. V dnešní době je velmi důležité najít co nejvhodnější řešení, které se liší podle nároků na něj kladených, jelikož zabezpečení se dosahuje zvýšením nadbytečnosti na úkor vlastního přenosu informace. Hledání alternativních řešení korekce shlukových chyb k dosud používaným metodám je v oblasti zájmu disertační práce.

Základním cílem práce je nalézt adekvátní náhradu hromadně používaných systémů pro opravu shluků chyb, jejichž nasazení v individuálních protichybových systémech umožní efektivnější přenos dat. Především odvodit podrobný matematický aparát pro rozšíření souboru kritérií, jež umožní lepší srovnání protichybových systémů vhodných pro přenos digitálních signálů. Simulacemi zvolených systémů ověřit získané teoretické výsledky. Následně výsledky využít k hledání přiměřených realizačních metod s ohledem na umístění do nadřazených přenosových systémů. Vzhledem k další využitelnosti získaných výsledků, zefektivňování procesu návrhu i realizaci číslicových systémů a snadné přenositelnosti je realizace demonstrovaná pomocí programově logických obvodů, jelikož i díky rekonfigurovatelnosti se v dnešní době hardware navrhuje i technikami dříve určenými jen pro návrh software.

2 MOŽNOSTI KOREKCE SHLUKOVÝCH CHYB

Přenos zpráv, který je na vstupu i výstupu příslušného přenosového systému nejčastěji dvoustavový, se v poslední době provádí převážně pomocí diskrétního signálu. Nebezpečí vzniklé chyby pro příjemce představuje především možnost nesprávné reakce systému na přenesenou zprávu. Zmíněný jev nastává tehdy, jestliže se informace přenášená sledovaným úsekem významově změní v jinou a zcela odlišnou část užívané zprávy. Stává se tak, když soubor možných kombinací signálových prvků v dané části zprávy je využíván beze zbytku jen pro přenos informace. Počet míst, ve kterých se dvě kombinace prvků ve sledovaném úseku zprávy mezi sebou liší se nazývá Hammingova vzdálenost *d*. Umělým zvyšováním nadbytečnosti přenášené posloupnosti signálových prvků se zvyšuje Hammingova vzdálenost, ovšem zároveň se snižuje hodnota přenášené informace.

Miniaturizace místa pro záznam dat a zrychlování přenosu dat neustále pokračuje v překotném tempu. Pro spolehlivou funkčnost je nepostradatelné se zabývat ochrannou před nežádoucími vlivy i tam, kde to dříve nebylo nutné [11]. Stále častěji se místo jednoduchých chyb či vícenásobných chyb vyskytují chyby shlukové. K jejich odstranění se používají odlišnější metody [5]. K potlačení shluků chyb či k opravě chybných bitů lze využít několika rozdílných metod. Stručná charakteristika a principy jsou představeny v následujících podkapitolách.

2.1 BLOKOVÉ KÓDY

Uskutečňují zabezpečovací proces pouze v rámci jediného bloku, který vznikl oddělením určitého úseku signálových prvků. Sledovaný úsek zprávy se nazývá kódové slovo. Při používání blokového kódování velmi rychle narůstá množství redundantních dat s počtem bitů, které má být kódování schopno opravit v daném bloku. Proto se tato metoda využívá zejména tam, kde nárůst datového toku není kritický. Především je uvedená skupina kódů zaměřena na opravu nezávislých chyb.

Výjimku mezi blokovými kódy ve schopnosti opravování chyb tvoří Reed-Solomonovy kódy, jenž patří mezi nebinární cyklické BCH kódy (Bose-Chaudhuri-Hocquenghem) a jejich abeceda zdroje je vždy vyšší než binární, proto se využívají pro korekci shlukových chyb. Označují se zkratkou RS *(n, k)*, která charakterizuje daný kód. Parametr *k* určuje počet *m* - bitových symbolů vstupujících do kodéru, parametr *n* udává velikost zprávy vystupující z kodéru. Kódování se neprovádí nad jednotlivými bity, ale nad symboly (byty). Nezáleží na tom, kolik chybných bitů obsahuje jeden symbol, ale pouze na tom, zda je chybný či nikoliv. Uvedené kódy nejsou efektivní pro opravu nezávislých, ojedinělých chyb [19]. Tuto nevýhodu RS kódů lze odstranit pomocí zřetězených kódů, jenž využívá dva kodéry zapojené v kaskádě [9], [20].

Donedávna byla softwarová realizace v reálném čase vzhledem ke složitosti výpočtů neuskutečnitelná. Aritmetické operace v konečných polích GF(2^m) jsou dosti odlišné od operací v binárním číselném systému. Komplikovaná je zejména realizace násobení a dělení. Tedy velké obtíže u zmíněné implementace jsou hlavně

z důvodu nepodporování Galoisova tělesa a k němu patřících aritmetických operací procesorem. Nicméně dnes dostupné výkonné procesory již umožňují zpracování dat velmi vysokými rychlostmi. Avšak stále složitost a výpočetní náročnost kódování je velmi vysoká, proto je mnohdy vhodnější používat systémy opravující menší shluky chyb, za to však s rychlejším kódováním přenášených dat.

2.2 PROKLÁDÁNÍ

Tvoří používaný doplněk kanálového kódování při předchozím využití jiného kódování proti chybám [7], [15]. Prokládání (interleaving) se používá jako ochrana signálu proti skupinovým chybám – shluku chyb. Prokládání zprávy představuje přídavný systém, který se vkládá ve vysílači mezi kodér a přenosový kanál a na straně příjemce mezi přenosový kanál a dekodér. Základní princip spočívá ve změně polohy bitových toků tak, aby se charakter distribuce shluku chyb změnil na distribuci nezávislých chyb, jež lze účinněji odstranit či potlačit. Nevýhoda této dodatečně prováděné operace spočívá v dalším zpožďování dekódování bitů na straně příjemce. Zpoždění způsobují prokládací matice a je určeno jejich velikostí jak v části vysílače, tak v části přijímače.

U blokové metody jsou kódové kombinace ukládány do paměti matice prokládání, jež má rozměr n x j tak, že v každém řádku je jedna kódová kombinace. Až se matice určená pro prokládání naplní, jsou z ní uložené bity vysílány do přenosového kanálu, ale tentokrát po jednotlivých sloupcích. Vzniklý bitový tok je přenášen přes přenosový kanál do stejné paměťové matice, kde je ukládán po sloupcích. Po naplnění paměťové matice jsou přenesené bity vysílány po řádcích do dekodéru příslušného korekčního kódu, kde jsou chybně přenesené bity opraveny. Ve sdělovacím kanále jsou každé dva bity původní kódové kombinace odděleny (j – 1) bity ostatních kódových kombinací. Je-li pak přenášený bitový tok napaden shlukem chyb, je vzhledem k použitému korekčnímu kódu napadeno všech j kódových kombinací, avšak j-krát menším počtem chyb. Signál s ojedinělými chybami lze snáze opravit a již může být aplikován vhodný korekční kód, který předtím neměl šanci provést úspěšnou korekci. Většinou se tento systém používá jako doplněk blokového kódování [2], jenž je účinné proti ojedinělým chybám.

Rozdíl mezi konvolučním a bitovým prokládáním je stejný jako rozdíl mezi blokovým a konvolučním kódováním [17]. U blokového prokládání probíhá práce po blocích, které jsou načítány do matice. U konvolučního prokládání celý mechanismus probíhá opět průběžně. Bitový tok, nejčastěji rozdělený do značek, se ukládá do jednotlivých paměťových větví konvolučního prokladače. Po naplnění všech větví n prokladače se vrací na začátek a v tu chvíli dochází k přímému propojení mezi vstupem a výstupem. Tím dochází k různému zpoždění značek a jejich vzájemnému promíchání při "nulovém" zpoždění mezi vstupem a výstupem.

2.3 KONVOLUČNÍ KÓDY

Konvoluční kodéry lze popisovat jako zdroje zpráv s pamětí s nepřetržitým způsobem zabezpečení. Způsob kódování určité informační posloupnosti závisí nejenom na aktuální vstupní informační posloupnosti, ale též na několika předchozích vstupních slovech [21]. Pro odvození výstupního kódového slova na straně vysílače lze dosáhnout daleko menšího nárůstu potřebného množství redundantních dat na zabezpečení, než je tomu u kódů blokových [28].

Na vstupu kodéru se rozdělí sériový tok na k_0 vstupních dílčích paralelních toků. V jedné větvi se uskuteční zabezpečující proces pomocí kombinačních obvodů. Na výstupu je n_0 dílčích výstupních posloupností, které se převedou zpět na sériový tok. Konvoluční kodéry se nejčastěji realizují posuvnými registry s odlišnou rychlostí posuvu [4]. V jednom kroku, kdy se vytvoří kódové slovo, dojde k posuvu na vstupu o k_0 zdrojových symbolů, ovšem na výstupu bude posuv o n_0 kódových symbolů. V dekodéru se vstupní data dělí na informační a zabezpečující část. K ověření korektnosti přijatých dat se opět počítají zabezpečující data. Dojde ke srovnání kontrolního bitu ze vstupu dekodéru a odpovídajícího vypočítaného bitu z posloupnosti v paměťovém poli dekodéru [23]. Pokud je zjištěn nesoulad vypočítává se poloha chyby, která je následně opravena. Dekódované slovo se získá až po uplynutí dopravního zpoždění, jež vzniká zpracováním úseku zprávy a případným použitím sériového přenosu zprávy.

Tato skupina kódů není tak známá, jako předchozí představené varianty, avšak v individuálních protichybových systémech může dosáhnout lepších výsledků [12] než výše uvedené v současnosti hojně používané metody. Na opomíjení konvolučních kódů lze usuzovat dle výskytu a dostupnosti literatury i technických zpráv pojednávající o dané problematice. Proto této problematice bude podrobněji věnována další část práce.

Turbokódy vychází z představy paralelně zřetězených kódů (vzájemně oddělených blokem prokládání) a iterativního způsobu dekódování. Základní myšlenka spočívá ve využití kombinace jednoduchých konvolučních kódů v paralelním zřetězení tak, aby každý z těchto kódů mohl být dekódován odděleně v méně složitém dekodéru. Turbokódy umožňují dosáhnout nízké chybovosti BER (Bit Error Rate) i při hodnotách SNR (Signal to Noise Ratio), jež leží velmi blízko Shannonovu limitu [8]. Ve většině případů se turbokodér skládá ze dvou (případně z více) paralelně zřetězených konvolučních kodérů. V praxi se nejčastěji využívají dva identické rekurzivní systematické konvoluční kodéry.

Prokladač představuje klíčový faktor, který ovlivňuje celkovou výkonnost turbokódů. Hlavním úkolem prokladačů v tomto případě je zajistit, společně s RSC kodéry, aby Hammingova váha výsledného kódového slova byla velká i v případech, kdy se na vstupu turbokodéru objeví některá z nejméně vhodných datových sekvencí. Složitost i náročnost je vysoká a celková informační rychlost příliš nízká. Vhodněji lze pro některé případy použít systémy opravující menší shluky chyb, za to však s rychlejším kódováním přenášených dat a docílit vyšší informační rychlosti.

3 PROTICHYBOVÉ KÓDOVÁNÍ

Chybovostí v systému se zabývá, až pokud výskyt různých typů chyb překročí únosnou mez. Shlukové chyby jsou v oblasti miniaturizace místa pro záznam dat a zrychlování přenosu dat problémem, který mění zásadním způsobem pohled na zabezpečení užitečné informace před nežádoucími účinky šumu [18]. Ukazuje se, že pro tuto skupinu chyb, již nejsou příliš vhodné systémy pro opravu náhodných chyb, které se momentálně hojně využívají [13]. V současnosti je problém shlukových chyb převážně řešen použitím blokových kódů s prokládáním či RS kódů. Záměrem je vytvoření rozšířené metodiky pro návrh individuálních protichybových systémů využívajících alternativního řešení korekce shlukových chyb pomocí vybraných konvolučních kódů. Rozšíření popisu systematických konvolučních kódů pomocí matematického aparátu umožní dokonalejší posuzování jednotlivých kódových zabezpečení i srovnání s dosud hromadně používanými univerzálními řešeními.

Při návrhu individuálních protichybových systémů musí konstrukce kódovacího a dekódovacího zařízení sledovat několik základních cílů: rychlé kódování informace, snadný přenos zakódované zprávy, rychlé dekódování přijaté zprávy, opravu chyb způsobených šumem v kanálu během přenosu zprávy, maximalizaci množství informace přenesené za jednotku času. Hlavním bodem je čtvrtý z těchto úkolů. Problém spočívá v tom, že dosažení čtvrtého cíle není v souladu s pátým cílem, a nemusí být ani příliš v souladu s prvními třemi uvedenými úkoly. Jakékoliv řešení tohoto problému je nutně kompromisem mezi těmito pěti cíly. Na základě zjištěných výsledků lze hledat variantní řešení pro již používané způsoby protichybových zabezpečení s ohledem na dosažení vyšší efektivity výsledků.

3.1 ŘEŠENÍ U INDIVIDUÁLNÍCH PROTICHYBOVÝCH SYSTÉMŮ

Vývoj kráčí dopředu především směrem speciálních robustních systémů se složitými výpočetními algoritmy a snaží se tak pokrýt většinu požadavků [3]. Avšak u individuálních protichybových systémů, kde důraz je kladen i na malou výpočetní náročnost, ať již kvůli ceně zařízení či kvůli většímu množství přenesených dat, tyto systémy nejsou vždy tím nejvhodnějším možným řešením. Jelikož jednodušší systémy mohou poskytnout v některých případech mnohem vyšší požadovanou efektivitu [12]. U individuálních protichybových systémů se vyhledávají specifická řešení, jenž dokáží poskytnout lepší vlastnosti než univerzální varianty.

U problematiky specifických řešení se analýza shlukových chyb opírá o možnosti modelování a následné simulace dějů kódování, dekódování a přenosu v síti za pomocí vhodných softwarových nástrojů [10], které především slouží k ověření navrženého teoretického zapojení. Hlavní zaměření disertační práce se zabývá zmíněnými problémy v této kapitole – rozšíření možnosti vyhledávání specifických řešení pro opravu shluků chyb pomocí systematických konvolučních kódů u individuálních protichybových systémů.

4 APLIKACE SYSTEMATICKÝCH KONVOLUČNÍCH KÓDŮ

Konvoluční kódy umožňují omezit vliv šumu, jenž způsobuje vznik shlukových chyb. Na rozdíl od kódů blokových dochází k daleko menšímu rozrůstání potřebného množství zabezpečujících dat [22] a tím je k dispozici větší přenosová informační rychlost. Využití interleavingu je vázáno na předchozí protichybové kódování a tvoří pouze další přídavný systém, jež navíc způsobuje zpožďování přenášené informace. Samostatné použití zmíněného systému nepřináší žádné zlepšení. Robustní systémy pro opravu shluků chyb představují univerzálnější řešení, mnohdy však jednoduchost systémů využívajících konvolučních kódů umožní efektivnější využití výpočetních prostředků a tím rychlejší přenos dat.

Pro vytváření konvolučních kódů se využívají dvě základní metody [5]. Kód lze zadat buď pomocí skupiny vytvářecích mnohočlenů, nebo pomocí vytvářecí matice. Obecnější je zápis pomocí vytvářecí matice, která je definována následovně:

$$\mathbf{F} = \mathbf{P} * \mathbf{G}, \tag{4.1}$$

kde řádky matice **P** představují dílčí vstupní toky, **F** dílčí výstupní toky a **G** je vytvářecí matice konvolučního kódu a je polonekonečná. Kontrola správnosti přenesených signálových prvků daného konvolučního kódování se provede vynásobením příchozí zprávy **F*** s kontrolní maticí **H**, kterou lze odvodit z vytvářecí matice **G**. Dva nejznámější dekódovací způsoby se rozdělují podle způsobů, jež využívají [16] na prahové a pravděpodobnostní.

4.1 ÚVOD K JEDNOTLIVÝM KÓDŮM

Poprvé byly představeny kódy pro opravu shlukových chyb Hagelbargerem. Nezávisle na sobě Iwadari a Massey zkonstruovaly efektivnější kódy stejného typu. Konvoluční kódy pro opravu postupných shlukových chyb byly studovány Wynerem a Ashem. Optimální kódy pro opravu postupných shlukových chyb byly později objeveny nezávisle Berlekampem a Preparatem.

Pro výše uvedené systematické konvoluční kódy bude ukázán návrh kodéru a dekodéru, který je schopen opravit shluk 4 chyb. Při této velikosti shluku chyb jsou kompletní schémata zapojení stále ještě dostatečně přehledná. Jednotlivé kódy jsou popsány především vytvářecí blokovou maticí $\mathbf{B_0}$. Odvozením matematického aparátu, jenž podrobněji charakterizuje dané kódy, lze hledat nejlepší variantu mezi touto skupinou kódů. Při využití získaných vztahů pro celkové zpoždění způsobené průchodem zprávy kodekem *Z*, celkové konstrukční složitosti kodeku S_L a S_P se zvýší počet srovnávacích kritérií. Vyhodnocením se dosáhne detailnějšího rozboru než jen při srovnání pomocí bezchybného intervalu *A*. Jelikož základ srovnání vychází u všech kódů ze stejného informačního poměru *R* a maximální opravitelné délky shluku chyb *b*.

4.2 ZÁKLADNÍ HAGELBARGERŮV KÓD (n_0; n_0 -1)

Kód lze určit pomocí blokové matice v dekadickém tvaru $\mathbf{B_D}$, která má tvar čtvercové matice rozměru n_0 x n_0 [24]. V každém řádku v místě odpovídající prvku diagonály je na úhlopříčce liché dekadické číslo. Pro $n_0 = 4$ platí vztah:

$$\mathbf{B_D} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 7 & 0 \\ 0 & 5 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix} \rightarrow \mathbf{B_0} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (4.2)$$

Matice $\mathbf{B_D}$ se pro potřebu binárních kódů převádí na blokovou matici v binárním tvaru $\mathbf{B_0}$. Počet míst dekadického čísla n, potřebných pro jeho zapsání dvojkovým číslem, se značí $L(n)$, což je nejmenší celé číslo, které vyhovuje nerovnosti:

$$L(n) \geq 1 + \log_2 n. \quad (4.3)$$

Binární čísla s menším počtem míst mají v matici $\mathbf{B_0}$ na nepoužitých místech nuly. První číslo odspodu, jenž je v dekadické podobě blokové matice rovno číslu tři, má vždy pouze dva řádky, protože nuly v místech nepoužitých dvojkových míst by způsobovaly v realizaci kodéru pouze neužitečné časové zpoždění, pak bloková matice $\mathbf{B_0}$ má rozměry:

$$\mathbf{B_0}[n_0; n_0 - 1] = ((n_0 - 1) * L(n) + 2; n_0). \quad (4.4)$$

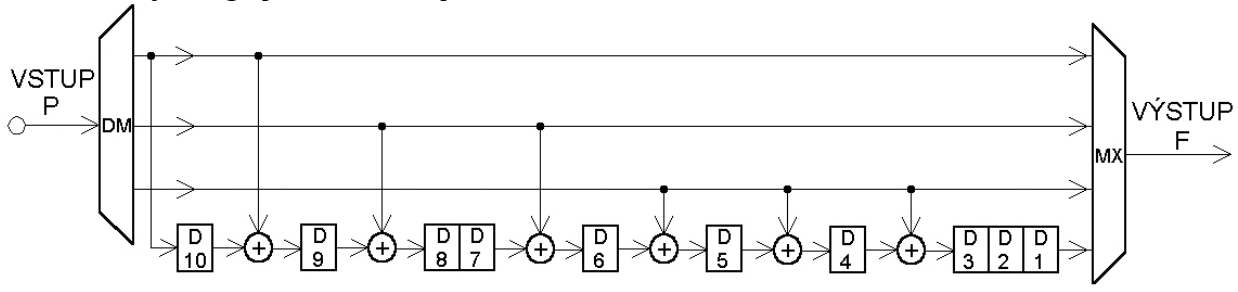
Tento kód opravuje shluky chyb délky b bitů:

$$b \leq n_0. \quad (4.5)$$

Mezi těmito shluky chyb musí být určitý bezchybný úsek – ochranný interval A:

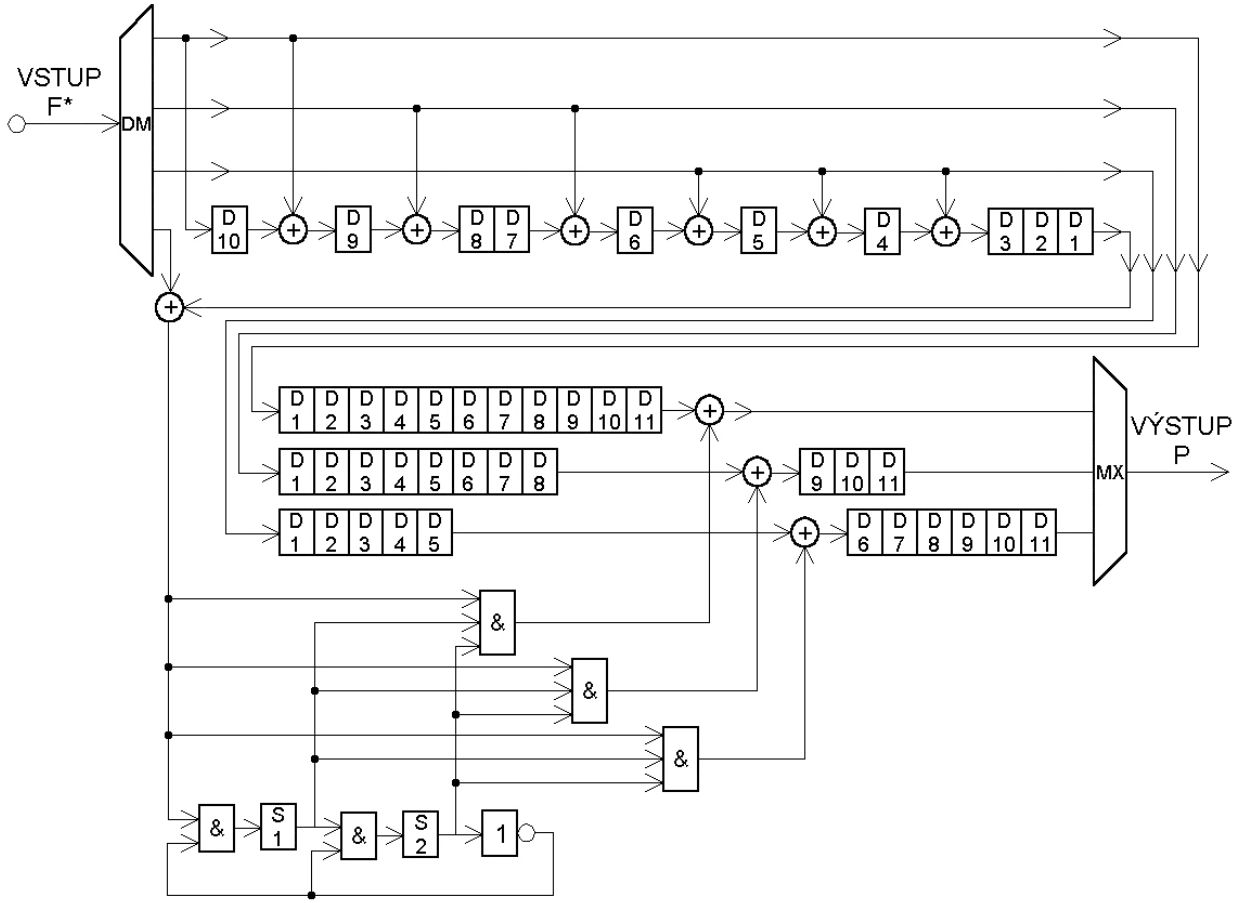
$$A \geq n_0^2 * L(n) - 1. \quad (4.6)$$

Vytvoření zapojení kodéru se odvozuje z $\mathbf{B_0}$ pomocí souboru vytvářecích mnohočlenů. Kódovací obvod je vytvořen tolika paměťovými buňkami, kolik řádků má $\mathbf{B_0}$. Mezi tyto buňky jsou zapojeny sčítačky mod2, dané vytvářecími mnohočleny. Zapojení kodéru jest uvedeno na Obr. 4.1.



Obr. 4.1: Kodér - Hagelbargerova kódu pro korekci 4 chyb.

Dekodér Hagelbargerova kódu využívá přímého převodu vektoru syndromu na chybový vektor (viz. Obr. 4.2). Z přeneseného bloku dat se oddělí zabezpečovací prvek, který se sečte mod2 s nově vypočítaným zabezpečovacím prvkem, jenž se vytvořil v kodéru přijímače. Tím vznikne syndromový prvek *s*, který slouží k nalezení a korekci chyb v dílčích posloupnostech přenesených informačních prvků.



Obr. 4.2: Dekodér Hagelbargerova kódu pro korekci 4 chyb.

S využitím získaných hodnot i na základě konkrétních schémat zapojení lze odvodit další parametry, jež lépe rozšíří popis daného systému [25]. Účelem je získat co nejobecnější vztahy, které budou mít všeobecnou platnost. Jejich využití jistě nalezne uplatnění při návrzích individuálních protichybových systémů, kdy bez předchozí podrobné znalosti daného systému se získá více popisných informací.

Základní zpoždění je způsobováno především průchodem bitů přes paměťové buňky. V kodéru dochází k průběžnému tvoření zabezpečujících bitů a použité paměťové buňky slouží jen k tomuto účelu. Přenášená informace tak zde není vůbec zpožďována. V dekodéru se avšak některé paměťové buňky používají k úschově přenesené informace za účelem případné korekce. Vlastní užitečná informace není ze vstupu dekodéru ihned k dispozici na výstupu dekodéru a již dochází ke zpoždění. Pro celkové základní zpoždění Hagelbargerova kodeku při paralelním přenosu platí vztah:

$$Z = Z_K + Z_D = Z_D = L(n) * n_0 - 1, \tag{4.7}$$

kde Z_K je zpoždění kodéru a Z_D zpoždění dekodéru. V případě sériového přenosu se v dekodéru uplatní jednotlivé větve, mezi které demultiplexor přijatý informační signál rozděluje:

$$Z_S = Z * (n_0 - 1) = L(n) * (n_0^2 - n_0) - n_0 + 1. \quad (4.8)$$

Z Obr. 4.1 a Obr. 4.2 je zřejmé, že pro realizaci zapojení se používají především dva typy prvků. Konstrukční složitost kodeku je dána zejména potřebným počtem paměťových buněk a logických operátorů. Multiplexory-demultiplexory na vstupu-výstupu systému netřeba uvažovat, jelikož ty zajišťují úpravu rychlosti zpracování, s čímž musí pracovat každý zabezpečovací kód. Pro počet paměťových buněk kodeku S_P, který se skládá z jednotlivých dílčích částí kodéru S_{PK} a dekodéru S_{PD}, po celém odvození platí vztah:

$$S_P = S_{PK} + S_{PD} = \left(L(n) * (n_0^2 + n_0 - 1)\right) - n_0 + 2. \quad (4.9)$$

Druhou skupinu tvoří jednotlivé logické operátory, jenž se podílejí na zajišťování vlastní ochrany přenášené informace. Pro počet logických operátorů kodeku S_L, který se skládá z jednotlivých dílčích částí kodéru S_{LK} a dekodéru S_{LD}, po celém odvození platí vztah:

$$S_L = S_{LK} + S_{LD} = 2(n_0 - 1) + 5(n_0 - 1) + 1 = 7(n_0 - 1) + 1. \quad (4.10)$$

4.3 ZÁKLADNÍ IWADARI-MASSEYHO KÓD ($m \cdot n_0$; $m \cdot k_0$)

Iwadari-Masseyho kód je popsán svojí vytvářecí blokovou maticí $\mathbf{B_0}$ [26], která má n_0 sloupců. Sloupce jsou indexovány zleva od a_i, řádky se číslují vzestupně zespodu. Na nejvyšším řádku úplně vpravo je umístěna jednička představující zabezpečovací prvek. Názorně vše uvádí vztah (4.11), kde je obecné schéma vytvářecí blokové matice $\mathbf{B_0}$ Iwadari-Masseyho kódu:

$$B_0 = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 \\ \vdots & & & & & \\ 0 & 1 & \dots & 0 & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ n_0 \text{ řádků} \\ \\ \\ \\ 2 \text{ řádky} \\ \\ 3 \text{ řádky} \\ \\ \\ n_0\text{-}1 \text{ řádků} \\ \\ \\ \\ n_0 \text{ řádků} \\ \\ \\ \end{matrix}$$

$$a_i \; b_i \qquad x_i \; y_i \; z_i \quad (4.11)$$

Bloková matice $\mathbf{B_0}$ má rozměry:

$$\mathbf{B_0}[m \cdot n_0; m \cdot k_0] = (m; n_0), \quad (4.12)$$

kde jednotlivé parametry kódu m a k_0 jsou definovány:

$$m = \frac{n_0 * (n_0 - 1)}{2} + (2n_0 - 1), \tag{4.13}$$

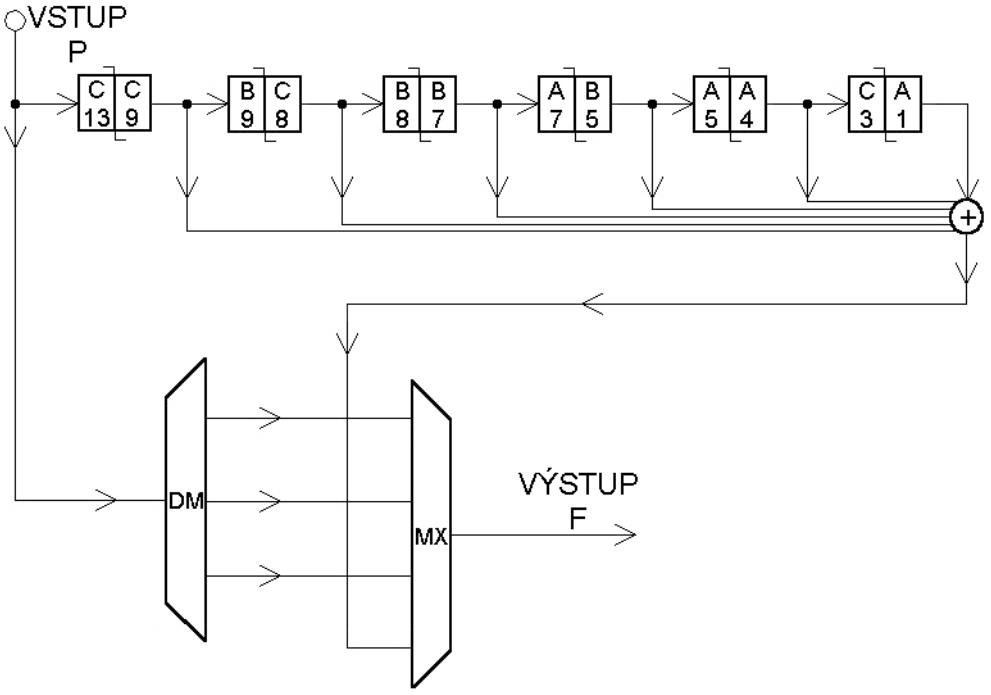
$$k_0 = n_0 - 1, \tag{4.14}$$

k_0 a n_0 představují počet dílčích vstupních či výstupních paralelních toků, m udává počet řádků příslušné vytvářecí matice. Pro parametry korekční schopnosti b a ochranného interval A platí:

$$b \le n_0, \tag{4.15}$$

$$A \ge n_0 * m - 1. \tag{4.16}$$

V případě překročení mezní zabezpečovací schopnosti nezpůsobuje tento kód nekonečný průnik chyby do vlastní informace [29]. Z vytvářecí blokové matice lze sestavit celý kodek Iwadariho kódu viz Obr. 4.3 a Obr. 4.4, kde vyobrazená schémata opět zabezpečují zprávu před shlukem chyb velikosti $b \le 4$. Základem je bloková vytvářecí matice $\mathbf{B_0}$ určená ze vztahu (4.11).



Obr. 4.3: Kodér Iwadari-Masseyho kódu pro korekci 4 chyb.

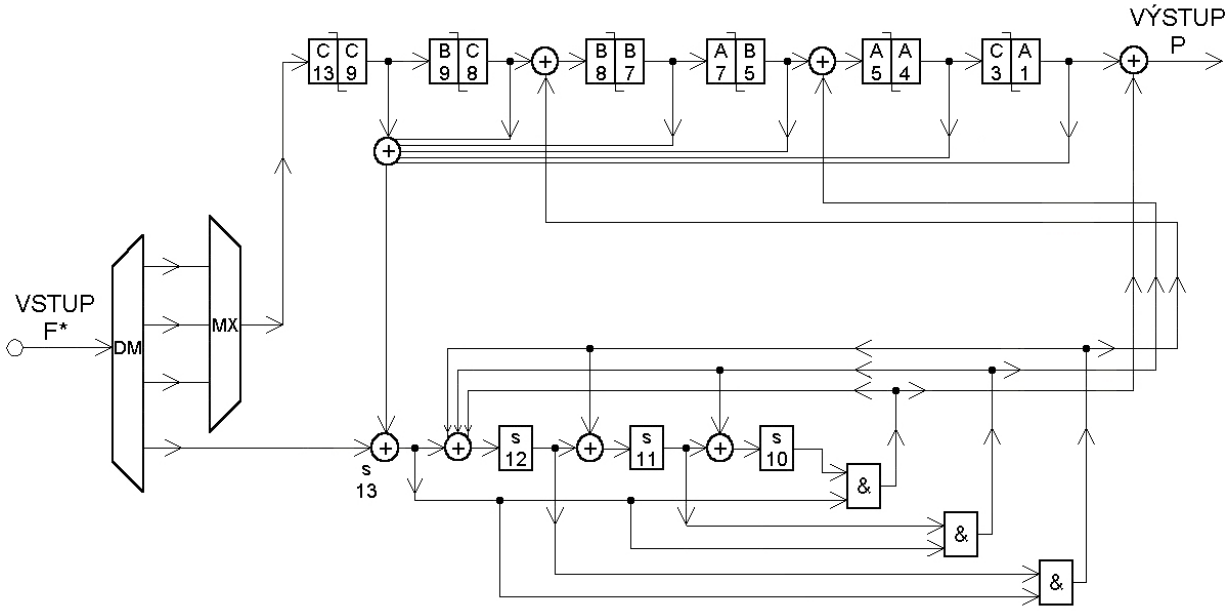
Způsob dekódování dat na straně příjemce je naznačen na Obr. 4.4. Princip dekodéru spočívá ve využití prahového dekódování. K syndromové rovnici pro určitý čas k bitům v ní obsažených je třeba nalézt bity na zabezpečení se podílejících v čase předcházejícím. Za pomocí dvou syndromových rovnic lze provést korekci jednoho určitého bitu.

S využitím získaných výsledků je možné provést hlubší rozbor kodeku, tak jako v předchozím případě [25]. Základní zpoždění je způsobeno jen průchodem přes paměťové buňky v dekodéru. Při paralelním způsobu odvozování i přenosu platí:

$$Z = Z_K + Z_D = Z_D = \frac{n_0 * (n_0 - 1)}{2} + (2n_0 - 1), \tag{4.17}$$

kde Z_K je zpoždění kodéru, Z_D zpoždění dekodéru. V případě sériového přenosu platí:

$$Z_S = m * k_0 = 0{,}5n_0^3 + n_0^2 - 2{,}5n_0 + 1. \quad (4.18)$$



Obr. 4.4: Dekodér Iwadari-Masseyho kódu pro korekci 4 chyb.

Při stanovení konstrukční složitosti se využívají znalosti rozměrů blokové matice $\mathbf{B_0}$ - stejně jako u Hagelbargerova kódu k vyjádření vztahů potřebného množství paměťových buněk a logických operátorů. Pro počet paměťových buněk kodeku S_P, který se skládá z jednotlivých dílčích částí kodéru S_{PK} a dekodéru S_{PD}, po celém odvození platí vztah:

$$S_P = S_{PK} + S_{PD} = (m * k_0) + ((m * k_0) + n_0 - 1) = n_0^3 + 2n_0^2 - 4n_0 + 1. \quad (4.19)$$

Zbývá určit množství potřebných logických operátorů. Pro počet logických operátorů kodeku S_L, který se skládá z jednotlivých dílčích částí kodéru S_{LK} a dekodéru S_{LD}, po celém odvození platí vztah:

$$S_L = S_{LK} + S_{LD} = 1 + 3(n_0 - 1) + 2 = 3n_0. \quad (4.20)$$

4.4 ZÁKLADNÍ BERLEKAMP-PREPARATŮV KÓD (n_0; n_0 -1; m)

Postup objevený nezávisle Berlekampem a Preparatem, vždy přinese kód splňující Gallagerovy meze. Gallager dokázal, že libovolný konvoluční kód o informační rychlosti R má schopnost opravit všechny shluky délky b nebo kratší vzhledem k ochrannému intervalu délky A, jestliže platí [22]:

$$\frac{A}{b} \geq \frac{1+R}{1-R}. \quad (4.21)$$

Jedná se o systematický konvoluční kód ke korekci shluků omezených do samostatného bloku úměrného ochrannému intervalu m bezchybných bloků. K základnímu popisu se používá vytvářecí matice $\mathbf{B_0}$, jejíž rozměry jsou n_0 x $2n_0$, jež se skládá ze dvou podmatic [32]. První podmatice má jedničky na vedlejší diagonále a druhá podmatice má jedničky umístěné nad hlavní diagonálou. Přičemž obě podmatice mají shodný rozměr n_0 x n_0. Zbylé prvky podmatic jsou nulové viz vztah:

$$\mathbf{B_0} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.22}$$

a obecná bloková matice $\mathbf{B_0}$ má rozměry:

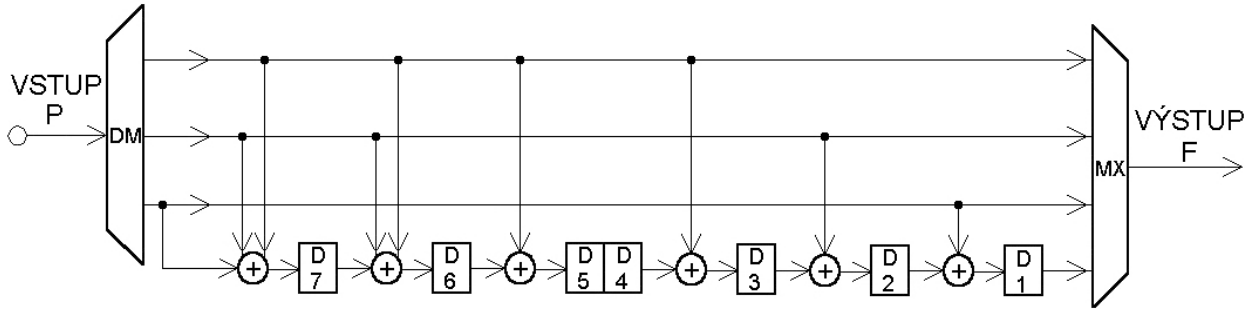
$$\mathbf{B_0}[n_0; n_0 - 1; m] = (n_0; 2n_0). \tag{4.23}$$

Pro další důležité parametry kódu platí:

$$A = m * n_0 = m * b, \tag{4.24}$$

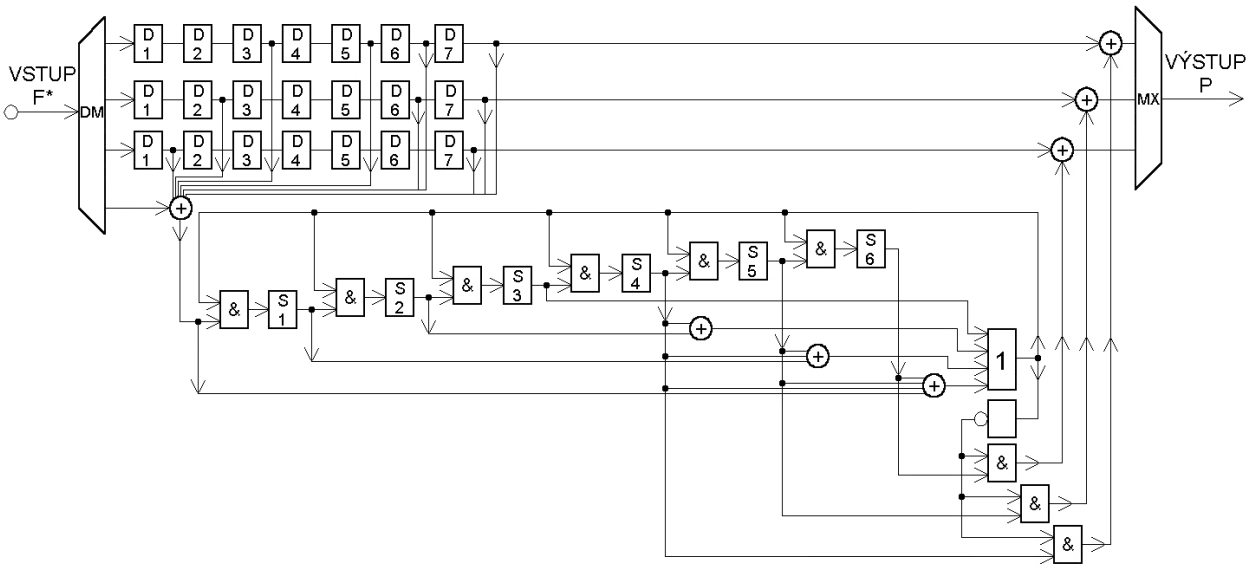
$$\frac{A}{b} = m = 2n_0 - 1. \tag{4.25}$$

Bloková vytvářecí matice $\mathbf{B_0}$ stanovuje způsob zapojení kodéru viz Obr. 4.5 – určují jej zabezpečovací vytvářecí mnohočleny.



Obr. 4.5: Kodér Berlekamp-Preparatova kódu pro korekci 4 chyb.

Berlekamp-Preparatovy kódy mohou být dekódovány použitím obecné dekódovací techniky pro konvoluční kódy opravující shlukové chyby zásluhou Masseyho [33]. V uvedeném dekodéru nemůže nastat nekonečné šíření chyby a jeho schéma je uvedeno na Obr. 4.6. Vychází se z kontrolní matice $\mathbf{H_0}$.



Obr. 4.6: Dekodér Berlekamp-Preparatova kódu pro korekci 4 chyb.

Základní zpoždění je způsobeno jen průchodem přes jednu dílčí část paralelního větvení paměťových buněk v dekodéru, jež slouží k vytvoření kodéru přijímače a využívá se jich pro úschovu přenesené informace za účelem případné korekce

chyb. V kodéru se zabezpečení tvoří průběžně, a tedy neovlivňuje přenášenou informaci. Pro základní zpoždění Berlekamp-Preparatova kodeku platí vztah:

$$Z = Z_K + Z_D = Z_D = 2n_0 - 1, \tag{4.26}$$

kde Z_K je zpoždění kodéru a Z_D zpoždění dekodéru. V případě sériového přenosu informace se v dekodéru uplatní zpoždění všech informačních větví:

$$Z_S = Z * (n_0 - 1) = 2n_0^2 - 3n_0 + 1. \tag{4.27}$$

Jako v dřívějších případech se využije známosti rozměrů blokové matice $\mathbf{B_0}$ pro stanovení konstrukční složitosti jednotlivých celků. Pro počet paměťových buněk kodeku S_P, který se skládá z jednotlivých dílčích částí kodéru S_{PK} a dekodéru S_{PD}, po celém odvození platí vztah:

$$S_P = S_{PK} + S_{PD} = (2n_0 - 1) + (2n_0^2 - n_0 - 1) = 2n_0^2 + n_0 - 2. \tag{4.28}$$

Závěrem pro počet logických operátorů kodeku S_L, který se skládá z jednotlivých dílčích částí kodéru S_{LK} a dekodéru S_{LD}, po celém odvození platí vztah:

$$S_L = S_{LK} + S_{LD} = (2n_0 - 2) + (5n_0 - 2) = 7n_0 - 4. \tag{4.29}$$

4.5 SROVNÁNÍ JEDNOTLIVÝCH VARIANT

Vlastnosti prezentovaných konvolučních kódů, jsou uvedeny v Tab. 4.1. Hodnoty doplňujících vlastností konvolučních kódů v tabulce jsou uvedeny pro maximální opravitelnou délku shluku chyb $b \leq 4$ pomocí jednotlivých vztahů z kapitoly 4.2, 4.3 a 4.4. Dalším společným znakem všech kódů je stejná informační rychlost R = 0,75.

Tab. 4.1: Konvoluční kódy s hodnotami sledovaných vlastností.

Kód	A [bit]	Z [bit]	S_P [-]	S_L [-]
Hagelbargerův	47	11	55	23
Iwadari-Masseyho	51	13	81	12
Berlekamp-Preparatův	28	7	34	24

Z tabulky je zřejmé, že při stejné informační rychlosti a velikosti opravitelné chyby má sledované veličiny nejnižší (a tím pádem nejlepší vlastnosti) Berlekamp-Preparatův kód. Vyžaduje nejkratší ochranný interval, způsobuje nejkratší zpoždění přenášené zprávy průchodem paměťových buněk a k realizaci kódu je třeba nejmenší počet paměťových buněk. Uvedené vlastnosti jsou vykoupeny největším počtem potřebných logických operátorů k zajištění činnosti kodeku. Však rozdíl není tak dramatický jako u počtu paměťových buněk. Nejméně logických funkcí pro správné zajištění funkčnosti kodeku dle Tab. 4.1 představuje Iwadari-Masseyho kód. Výše popsaný způsob pracuje s kódovým zabezpečením v podstatně širších souvislostech. Neomezuje se pouze na zabezpečovací kód a všímá si dalších vlastností, které jsou důležité pro výběr optimální varianty při realizaci [24], [25].

Konfrontace nejčastěji používaných protichybových systémů [1], [3] se systémem konvolučního kódování je provedena pro shluk chyb délky $b \leq 5$ bitů. Lze se domnívat, že pro klasické blokové kódy velikost této chyby již představuje zvýšené

nároky na zabezpečení jednoho bloku a dochází tak ke snížení dosahované informační rychlosti. Při nasazení robustních systémů sloužících k opravě dlouhých shluků chyb se stále ještě jedná o plýtvání jejich výpočetním výkonem, jelikož využívají komplikovaných výpočetních algoritmů jak pro kódování tak především pro dekódování. U individuálních protichybových systémů, kde je vyžadováno při konstrukci především splnění všech protichůdných podmínek a neklade se důraz na masovou výrobu, lze pak v některých oblastech dosáhnout lepších výsledků [3], [12], [36].

Hammingův kód je vybrán pro svoji jednoduchost a velmi nenáročný způsob kódování i dekódování. Což jsou vlastnosti, které mu při hledání nejlepších variant řešení opravy náhodných chyb poskytují vysoké ohodnocení [6], [12]. Jelikož Hammingův kód (7, 4) je schopen opravit pouze jedinou chybu je třeba pro opravu chybného úseku 5 bitů využít navíc systém založený na bitovém prokládání, jenž zaručí rozložení shluku chyb do jednotlivých chyb v každém bloku. Pro rozměry požadované prokládací matice platí následující parametry:

$$j \geq \frac{b}{t} \geq \frac{5}{1} \geq 5,$$

pak velikost prokládací matice je dána:

$$PM = j * n = 5 * 7 = 35 bitů.$$

Kódové slovo má délku $n = 7$ bitů. Pro délku ochranného intervalu A:

$$A \geq n * j - b \geq 7 * 5 - 5 \geq 30 bitů.$$

Pro informační rychlost R platí:

$$R = \frac{k}{n} = \frac{4}{7} = 0{,}57.$$

Minimální zpoždění systému je dáno průchodem dvou prokládacích matic a velikosti informace ve zpracovávaném bloku.

$$Z_S = 2 * PM + k = 2 * j * n + k = 2 * 5 * 7 + 4 = 74 bitů.$$

Na stejný požadavek korekce shluku chyb $b \leq 5$ bitů se v druhém případě uplatní vícestavové kódy používané pro zabezpečení proti shlukům chyb – RS kódy. V tomto případě nebude nutné použít blok prokládání, jenž by jen zvyšoval celkové zpoždění. Jen počet opravovaných symbolů v daném bloku musí pokrýt počet chybných bitů – při libovolném umístění shlukové chyby. V případě použití čtyřstavových bitových symbolů platí:

$$n = 2^m - 1 = 2^4 - 1 = 15.$$

Pro počet chybných symbolů v kódu:

$$t \geq \frac{b}{m} \geq \frac{5}{4}, t = 2.$$

Tedy jeden shluk chyb poškodí dva čtyřstavové symboly bez ohledu na bitové poloze daného shluku. Počet informačních symbolů je roven:

$$k = n - 2t = 15 - 4 = 11.$$

Pak je definován kód RS (15, 11), který má informační rychlost R:

$$R = \frac{k}{n} = \frac{11}{15} = 0{,}73.$$

Shluk 5 chyb se může vyskytnout pouze jednou v daném bloku kódu RS (15, 11), aby jej bylo možné korigovat, a s tím souvisí i délka ochranného intervalu A:

$$A \geq n * m - b \geq 15 * 4 - 5 \geq 55 bitů.$$

Minimální zpoždění systému je dáno zpožděním doručení celého jednoho zpracovávaného informačního bloku:

$$Z_S = m * k = 4 * 11 = 44 bitů.$$

Na závěr se použije Berlekamp-Preparatův kód, což je zástupce konvolučních kódů pro opravu shlukových chyb. Ani v tomto případě nebude nutné použít blok prokládání, jenž by jen zvyšoval celkové zpoždění systému a navíc je možné navrhnout systém přesně na opravu daného shluku chyb. Dle vztahu (4.25):

$$b = n_0 = 5, m = 2n_0 - 1 = 10 - 1 = 9.$$

Definován je Berlekamp-Preparatův kód (5, 4, 9), který má informační rychlost R:

$$R = \frac{k_0}{n_0} = \frac{4}{5} = 0{,}80.$$

Délka ochranného intervalu A je jedním ze základních parametrů této skupiny kódů a pro její velikost platí (4.24):

$$A \geq m * n_0 \geq 9 * 5 \geq 45 bitů.$$

Minimální zpoždění systému při sériovém způsobu přenosu je dána vztahem (4.27):

$$Z_S = 2n_0^2 - 3n_0 + 1 = 2 * 5^2 + 3 * 5 + 1 = 36 bitů.$$

Všechny vypočítané vlastnosti představených zástupců jednotlivých používaných systémů pro korekci shluků chyb přehledně udává souhrnná **Tab. 4.2**. Výsledky potvrzují, že v individuálních protichybových systémech lze dosáhnout lepších parametrů, využitím skupiny konvolučních kódů pro opravu shluků chyb konkrétní velikosti.

Tab. 4.2: Hodnoty základních vlastností srovnávaných kódů pro $b \leq 5$.

Kód	A [bit]	Z_S [bit]	R [-]
Hammingův (7, 4)	30	74	0,57
Reed-Solomonův (15, 11)	55	44	0,73
Berlekamp-Preparatův (5, 4, 9)	45	36	0,80

5 SIMULACE

Pro prověření správnosti konstrukce schémat a zabezpečovacích schopností jednotlivých konvolučních kódů jsou získané výsledky podrobeny kontrole pomocí simulace. Simulace představuje základní metodu pro ověření i prezentování již navrženého zabezpečovacího procesu. Základní požadavky kladené na program jsou: stabilita, rozsáhlá knihovna prvků, uživatelsky přívětivé grafické prostředí, jednoduché a intuitivní ovládání, analýza případných chyb, podrobná dokumentace.

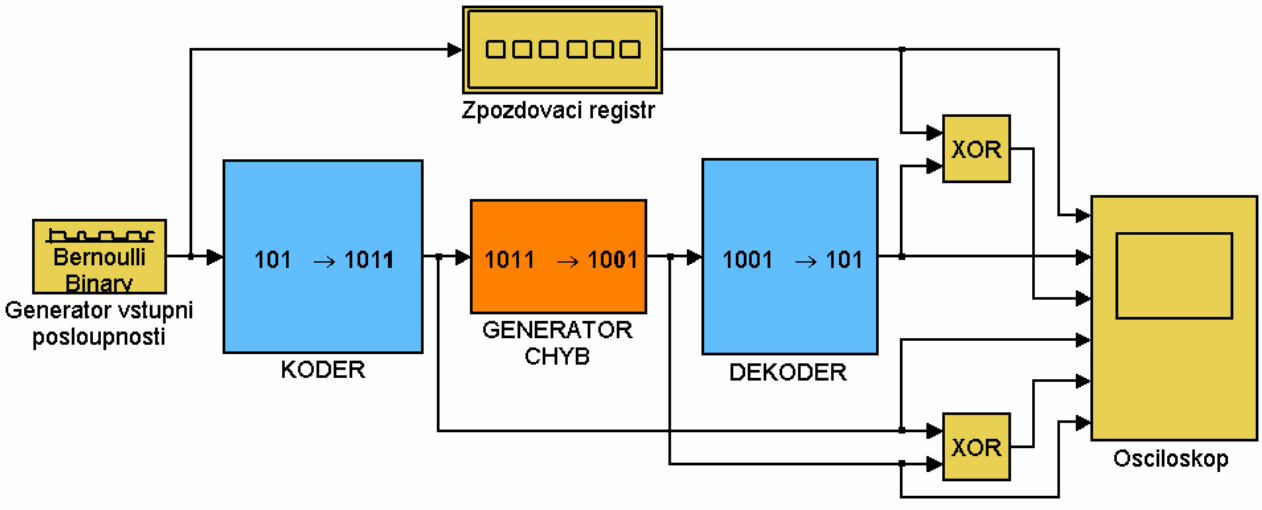
Programů pro simulaci je možno nalézt celou řadu - WinSpice, Micro-Cap, nadstavba Matlabu Simulink. Proto byl další z požadavků hlavně kladen na jednoduché a intuitivní ovládání. Za vhodnou volbu pro kódovací proces byl zvolen

matematický program Matlab [10], [27], [31]. Především lze využít specializované knihovny Simulink, jenž slouží k modelování a simulaci dynamických systémů. Graficky zadávaná soustava je složena z bloků, které jsou vybírány z různých knihoven. Prostředí Simulinku potom umožňuje graficky sledovat průběhy veličin v libovolném místě zapojení.

5.1 POPIS SIMULAČNÍHO ZAPOJENÍ MODELU KODEKU

Na Obr. 5.1 je uvedeno obecné blokové schéma zapojení daného simulovaného kodeku. Kromě bloku kodér a dekodér příslušného použitého kódu, jsou ještě použity bloky generátor vstupního toku dat, přenosový kanál (kde je umístěn generátor chyb) a zobrazení výstupního toku dat.

Bernoulliho binární generátor vytváří sériový nezabezpečený bitový tok, který jde na vstup demultiplexoru zvoleného kodéru. Na výstupu demultiplexoru kodéru jsou dílčí paralelní toky, jejichž počet je dán požadovaným zabezpečením, jež nezměněny projdou vlastním kodérem. Na výstupu příslušného kodéru se k nim přidá zabezpečující bitový tok, který byl v kodéru vytvořen. Všechny tyto bitové toky tvoří vstupy multiplexoru kodéru. Na výstupu multiplexoru kodéru je zabezpečený sériový bitový tok, jenž projde generátorem chyb, což představuje chybový přenosový kanál, který zavádí do přenášené posloupnosti shluky chyb. Na vstup demultiplexoru dekodéru se dostává zabezpečený sériový bitový tok se shluky chyb, jenž se rozdělí na paralelní bitové toky. Uvedené bitové toky tvoří vstupy vlastního dekodéru, kde se provádí případná korekce chyb, a na výstupu se dostávají původní dílčí paralelní bitové toky před přenosem. Na výstupu multiplexoru v dekodéru je původní, avšak zpožděný sériový bitový tok, jenž je vytvářen Bernoulliho binárním generátorem.



Obr. 5.1: Obecné blokové schéma zapojení kodeku.

Kodér se skládá ze tří dílčích podbloků. Prvním z nich je sérioparalelní převodník, který převádí vstupní sériová data na požadované paralelní toky dat pro kodér. Ve vlastním kodéru jsou vstupní data vybavena zabezpečovacím bitem dle pravidel daného kódu a pokračují do bloku paralelně sériového převodníku, který převádí

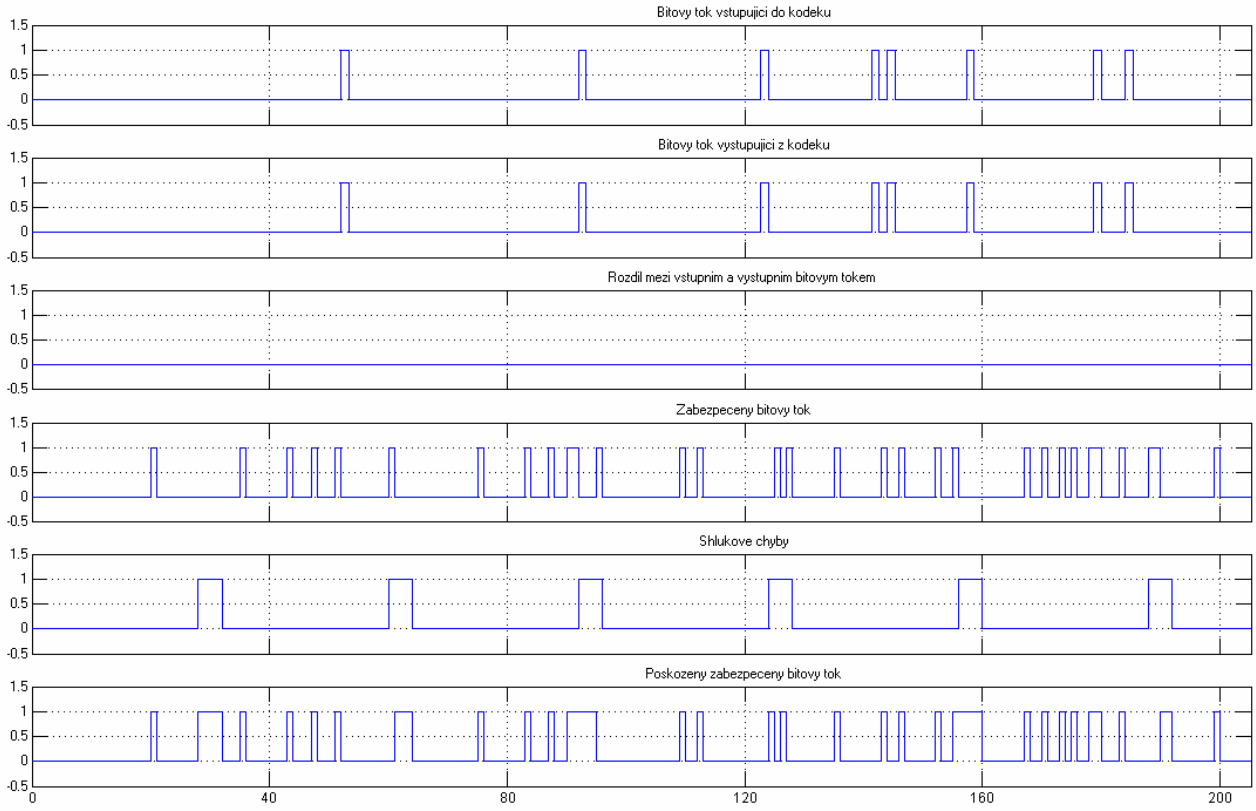
paralelní data zpět na sériový tok vhodný pro vysílání přenosovým kanálem. Pro sériově-paralelní a paralelně-sériové převodníky nelze použít bloky z knihovny Commonly Used Blocks – Mux, Demux, jelikož nejsou pro zadanou simulaci vhodné, protože u nich nelze nastavit dobu trvání jednoho vzorku vstupního a výstupního signálu [34]. Obrázky a popis jednotlivých bloků je rozveden ve vlastní práci.

Veškerá upravená - zabezpečená data jsou vyslána po přenosovém kanálu, kde může dojít k poškození. Na práci v binární soustavě je založen celý aparát detekce i korekce vzniklých chyb. Digitální přenos představuje posloupnost bitů, což jsou úseky datového přenosu o konstantní délce a určité úrovni, které v případě použití binárního kódu mají dvě hodnoty, a to 0 nebo 1. Z toho vyplývá i způsob vzniku chyb při přenosu takové posloupnosti nul a jedniček. Tedy k chybě v datovém kanálu může dojít různými vlivy pouze změnou hodnoty bitu, buď z 0 na 1, nebo z 1 na 0. Blok generátor chyb simuluje shlukové chyby, jež mohou vznikat v reálném přenosovém kanálu. Chybu lze modelovat jednoduše sčítačkou mod2, kdy je přičítán chybový vektor k přenášené zprávě nebo sofistikovaněji blokem generátoru chyb.

Dekodér je opět složen ze tří podbloků. Sériově paralelní převodník převádí vstupní data na dílčí paralelní toky. Vnitřní schéma zapojení dekodérů odpovídá příslušnému použitému konvolučnímu kódu viz Obr 4.2, Obr. 4.4 a Obr. 4.6. Jednotlivé opravené dílčí toky jsou na závěr pomocí paralelně sériového převodníku multiplexovány do výstupního bitového toku. Opět obrázky a popis jednotlivých bloků je podrobně rozveden ve vlastní práci.

Propojením všech dílčích částí lze provést kompletní simulaci kodeku daného modelu s grafickým výstupem viz Obr. 5.2. Vyobrazeny jsou následující grafy kodeku Berlekamp – Preparatova kódu pro opravu shluků chyb $b \leq 4$. Na prvním z nich je zobrazena vstupní nezabezpečená posloupnost bitů o přenosové rychlosti v_1, která je ovšem zpožděná. Zpoždění je zavedeno pro lepší přehlednost grafu (odpovídající signálové prvky jsou pod sebou). Na druhém grafu je zobrazena výstupní dekódovaná posloupnost bitů o stejné přenosové rychlosti. Třetí graf znázorňuje rozdíl mezi vstupními a dekódovanými daty. Nulový průběh značí shodu mezi signály. Na čtvrtém grafu jsou znázorněna data zakódovaná kodérem, jenž vstupují do simulovaného přenosového kanálu přenosovou rychlostí v_2. Následující graf ukazuje shlukové chyby, které ovlivňují přenášený signál. V posledním grafu jsou přenášená data s vloženými shluky chyb, jenž vstupující do obvodu dekodéru. Jelikož je zpožděná vstupní datová posloupnost rovna výstupní (viz třetí průběh) došlo tedy ke korekci vzniklých chyb.

Získané výsledky simulace potvrzují správnou funkci příslušného kodeku. Navržená schémata zapojení z kapitoly 4.2, 4.3, 4.4 a rozšiřující kritéria popisu kódů odvozená i na základě blokové vytvářecí matice byla touto simulací dostatečně prověřena [30], [35]. Toto grafické zobrazení poskytuje názornější pohled na problematiku zabezpečovacích korekčních kódů. Na přiloženém CD se nacházejí další detailní modely představených kodeků v kapitole 4, jejichž popis činnosti byl zde uveden.



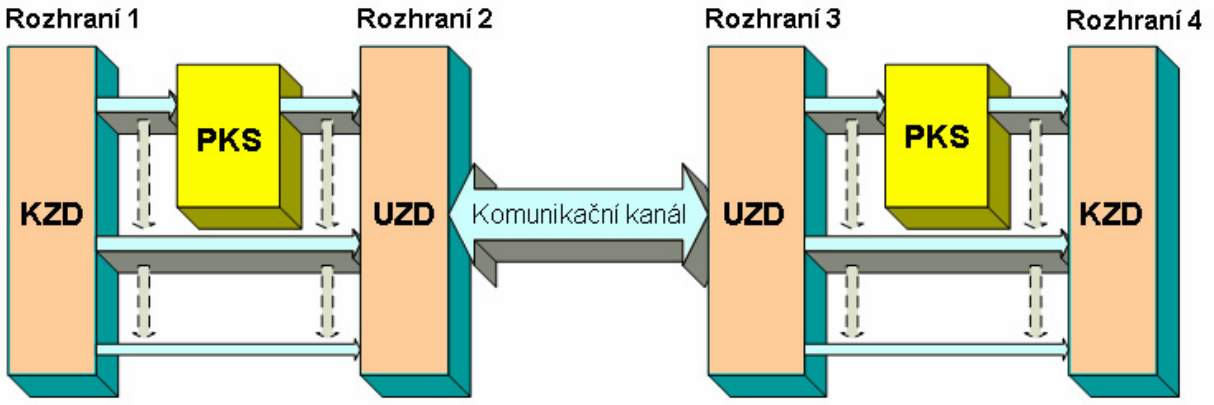
Obr. 5.2: Grafický výstup simulace pro Berlekamp – Preparatův kód.

6 REALIZACE

Zbývá ověřit realizovatelnost individuálních protichybových systémů. V tomto kroku je třeba součinnost zařízení nejen pro bezprostřední realizaci kódu. Proto začátek bude věnován začleněním vlastního protichybového systému v přenosovém řetězci. Následně se přistoupí ke způsobům návrhu číslicových systémů. Jelikož v posledních desetiletích se dramaticky zvýšila využitelnost číslicových systémů v nejrůznějších oborech, a tudíž je snaha co nejvíce zefektivnit i zlevnit proces jejich návrhu a realizace [14]. Obecně existuje řada způsobů návrhu číslicových systémů: slovní popis, matematický popis, obvodové schéma, programovací jazyk.

6.1 UMÍSTĚNÍ PROTICHYBOVÉHO KÓDOVÉHO SYSTÉMU

Je třeba brát ohled i na ostatní zařízení, která budou společně s kodekem tvořit příslušný zabezpečovací systém daného protichybového kódového systému. Ve své podstatě zahrnuje protichybový kódový systém nejen samotné kódovací a dekódovací algoritmy (procesy), ale pracuje s dalšími zařízeními potřebnými pro realizaci kódu a mimo to obsahuje i řadu dalších zařízení zajišťujících správnou činnost těchto dílčích částí viz [16]. PKS (protichybový kódový systém) je součástí NPS (nadřazeného přenosového systému). Umisťuje se v blízkosti vstupních prvků na vysílací straně a výstupních prvků na přijímací straně nadřazeného přenosového systému. Toto začlenění je patrné z Obr. 6.1.



Obr. 6.1: PKS umístěný v NPS.

6.2 SYNCHRONIZACE KODÉRU A DEKODÉRU

Obecně na všechny PKS je kladen požadavek na alespoň dvě přenosové rychlosti a tedy i na různou synchronizační frekvenci určitých prvků. Existují dvě základní možnosti, jak řešit problém s požadavkem na různé přenosové rychlosti [2]:

- Vstupní tok bitů protichybového kódového systému rozčlenit na přerušované úseky - pakety.
- V případě zachování nepřetržitého sériového toku informací se v systému vyskytnou minimálně dvě různé přenosové rychlosti. Jednotlivé prvky zabezpečovacího systému je třeba synchronizovat, respektive řídit odlišnými frekvencemi.

6.3 SOUBOR KRITERIÍ PRO VÝBĚR NEJVHODNĚJŠÍ VARIANTY

Soubor základních kriterií pro výběr nejvhodnější metody pro realizaci kodeku v individuálních protichybových systémech obsahuje požadavky z několika různých oblastí. Pro srovnání jednotlivých variant se zpravidla používá převod na normovanou veličinu [6], [16]. Výběr nejvhodnější varianty hardwarové implementace lze posuzovat dle následujících hledisek: cena realizace, finanční nároky, způsob popisu a realizace, nároky na napájecí napětí, možnost rozšiřitelnosti a modifikovatelnosti řešení, integrace řešení, náročnost realizace kodeku.

Na základě souboru kriterií lze učinit výběr vhodné metody pro hardwarovou implementaci, jenž může být znatelně ovlivněna přepočtem pomocí koeficientů, které se však mohou významně odlišovat u jednotlivých individuálních protichybových kódových systémů. Záleží především, na které vlastnosti se klade hlavní důraz, jelikož mnohé požadavky vystupují proti sobě a je třeba hledat kompromis.

6.4 VHDL

Vzhledem k uvedeným kritériím, perspektivnosti, využitelnosti se jeví nasazení FPGA obvodů pro individuální protichybové systémy jako správná volba [1], [6],

[12]. Pro konstruktivní simulaci, syntézu komponent a realizaci představuje charakterizace pomocí některého jazyka pro popis hardwaru vhodný výběr. Tyto jazyky se označují jako HDL a mezi známější patří jazyk VHDL, jenž byl velmi brzy přijat i jako standart mezinárodní organizace IEEE. Standardizování VHDL mimo jiné zapříčinilo rozšíření jazyka a jeho velkou přenositelnost, což představuje podstatnou výhodu při návrhu systémů vlastní realizace.

Při využití vypracovaných studií a simulací konvolučních kodérů a dekodérů pro digitální přenos signálů [27], [29], [35], se jako další vhodný krok nabízí pokračování v popisu pomocí jazyka VHDL, který je univerzální a přenositelný. Analýza na této úrovni představuje vlastní systémy realizace protichybového kódování, jelikož takto vytvořený systém je již možné použít a zařadit do nadřazených přenosových struktur. Tím bude dokončen rozbor rozšířeného popisu celého aparátu konvolučních kódů, jenž lze využít v individuálních protichybových systémech pro korekci shluků chyb.

Teoreticky odvozené obvodové schéma je ideální pro přímý přepis do strukturálního kódu ve VHDL [28]. Před vlastní implementací je ovšem potřeba navrhnout řešení několika dílčích problémů: jak popsat opakující se strukturu danou dle vytvářecí blokové matice, jak přesně použít multiplexory a demultiplexory ve funkcích sériově-paralelních a paralelně-sériových převodníků a také jak řešit problém odlišnosti vstupní a výstupní rychlosti komponent.

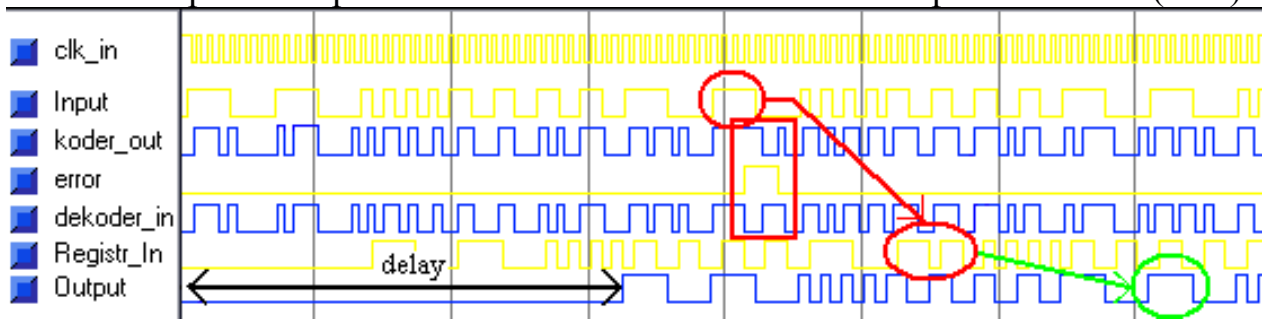
Opakující se struktura se skládá z navzájem propojených registrů (paměťových buněk) a logických členů XOR, jejichž počet a velikost u registrů je odvoditelná z parametru k_0. U strukturálního popisu komponent lze využít pro popis opakování příkaz GENERATE, což je jistá forma makra. K popisu určitých odlišností jednotlivých struktur generovaných pomocí uvedeného příkazu, se využije řídící proměnná tohoto počítaného cyklu.

V komponentách kodéru a dekodéru je potřeba různých synchronizačních signálů, jelikož jsou navrženy pro nepřerušovaný přenos dat. Potřebné jsou synchronizace pro práci se zabezpečeným tokem dat - to se týká prvků generujících výstup kodéru, zpracovávajících vstup dekodéru a s nezabezpečeným, čistě informačním, tokem dat - to se týká prvků zpracovávajících vstup kodéru či tvořících výstup dekodéru. Navíc se ještě využije synchronizace určitých prvků pro práci se syndromy a korekci bitů. Jelikož k přenosu zabezpečujícího bitu, výpočtu syndromu chyby i k případné korekci chyb v dekodéru dochází vždy po přenesení k_0 informačních bitů.

Přizpůsobení vstupu a výstupu kodéru i dekodéru je v obvodové realizaci navrženo použitím multiplexoru a demultiplexoru ve funkcích sériově-paralelního a paralelně-sériového převodníku [26]. Podrobnější analýzou je ovšem při realizaci možné najít efektivnější řešení. U vstupů obou komponent je dostačující, aby prvky pracující se vstupem jej vzorkovali s odpovídající frekvencí - navržená komponenta blok řízení. Pak zpožďovací posuvné registry kodéru načítají bity ze vstupu s frekvencí určenou signálem CLK_KODER_IN, se stejnou frekvencí načítají vstup zpožďovacích posuvných registrů dekodéru a s frekvencí CLK_DEKODER_SYN načítají vstup zpožďovací syndromové buňky dekodéru.

V produktu Modelsim XE III 6.3c byla provedena simulace popsaných komponent za účelem verifikace modelovaných komponent - na odpovídající vstupy se dostanou očekávané výstupy. Ovšem též slouží i pro jejich validaci, především u cílových komponent kodéru a dekodéru - ověřování zda celá funkčnost modelu odpovídá původním neformálním popisům funkce. Na základě zobrazených výstupů je možné posoudit očekávanost chování - verifikovat model.

Diagram signálových průběhů zobrazený s větším rozlišením je pro názornost na Obr. 6.2. Znázorněn je vstup a výstup kodéru (Input, koder_out), chybový vektor (error) znehodnocující vstup dekodéru (dekoder_in), poškozený bitový informační tok vstupující do posuvného registru (Registr_in) a opravený zpožděný bitový informační tok vystupující z posuvného registru dekodéru (Output). V diagramu je rámečkem znázorněn průnik chyby do přenosu. Ovály s šipkou v blízkosti zvýrazněného rámečku znázorňují výskyt shlukové chyby délky čtyř bitů, oprava shluku je znázorněna šipkou i oválem, jež je nejvíce vpravo a šipka nejvíce vlevo naznačuje informační zpoždění způsobené soustavou kodér-dekodér a odpovídá vztahu (4.18).



Obr. 6.2: Diagram signálových toků při korekci bitů.

Kompletní diagramy signálových toků do a z komponent kodér a dekodér, chování při nedodržení požadovaného ochranného intervalu *A* či překročení zabezpečovací schopnosti kódu jsou uvedeny i podrobně popsány ve vlastní práci.

Používání speciálních bloků příkazů takzvaných procesů je typické pro behaviorální styl popisu. Pro správné rozdělení příkazů do procesů je třeba odlišit, v závislosti na kterých vstupech a signálech vnitřního stavu systému se mění výstupy a vnitřní stav popisovaného systému. Stejně jako u strukturálního popisu je třeba navrhnout řešení popisu opakujících se struktur a popis různých synchronizací pro vstupy a výstupy kodeku navíc i logické členy XOR a AND, posuvné registry, zpožďovací buňky a převodníky signálů o různé synchronizaci.

Čítače při behaviorálním popisu lze využít i přímo v popisu funkce kodéru či dekodéru, kde na základě hodnoty těchto čítačů se propouští vstup do určitých prvků, popřípadě propouští výstup z určitých prvků na výstup celé komponenty. Pro periody potřebných signálů vyjádřené v počtech period dělené synchronizace platí:

$$T_{IN\_koder} = \frac{1}{f_{IN\_koder}} = \frac{t_B}{k_0} = \frac{k_0 * (k_0 + 1)}{k_0} * T_{divided} = (k_0 + 1) * T_{divided}. \qquad (6.1)$$

Pro výstup kodéru:

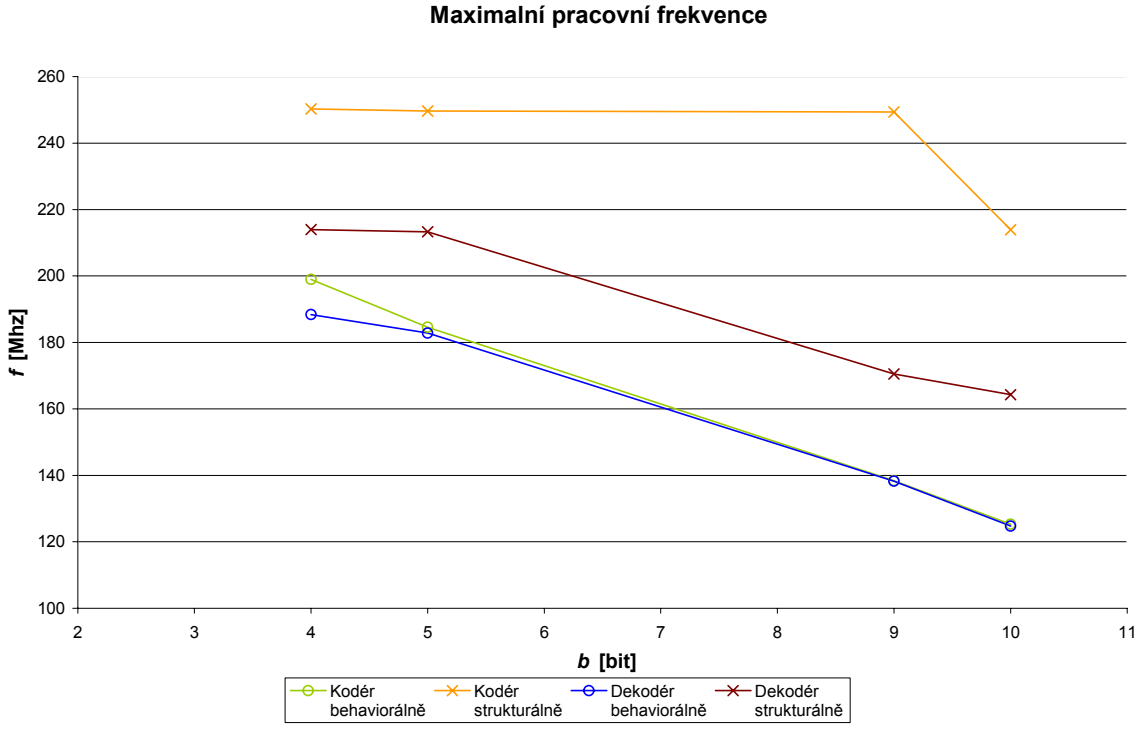
$$T_{OUT\_koder} = \frac{1}{f_{OUT\_koder}} = \frac{t_B}{k_0 + 1} = \frac{k_0 * (k_0 + 1)}{k_0 + 1} * T_{divided} = k_0 * T_{divided}. \qquad (6.2)$$

Pro prvky pracující se zabezpečením:

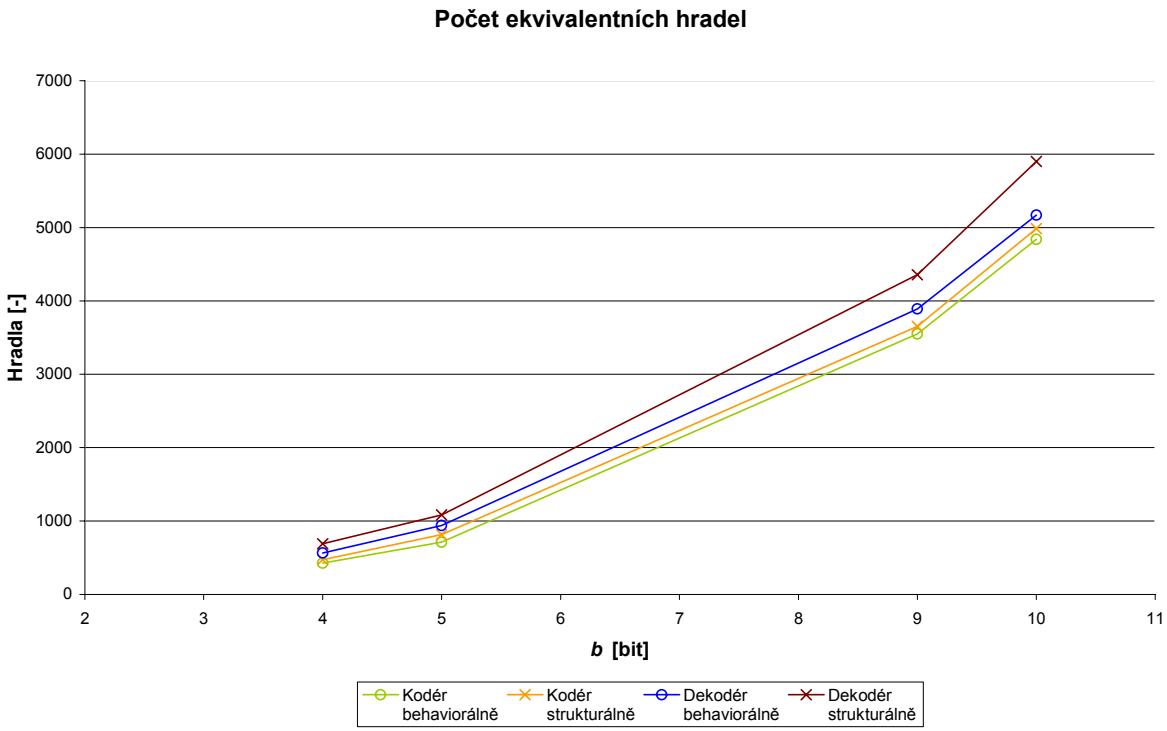
$$T_{SYN\_dekoder} = \frac{1}{f_{SYN\_dekoder}} = \frac{t_B}{1} = \frac{k_0 * (k_0 + 1)}{1} * T_{divided} = k_0 * (k_0 + 1) * T_{divided}. \qquad (6.3)$$

Příslušné synchronizační (respektive řídící) signály se mění vždy po vypočteném počtu změn signálu dělené synchronizace. Hodnoty čítačů řídí funkci celého kodéru i dekodéru. Takto je provedena plná náhrada funkce prvku PARALEL_TO_SERIAL, který zmíněné činnosti řídil v kodéru či dekodéru popsaném strukturálně. Analogické je testování behaviorálně popsaných komponent s výše uvedeným postupem jako u strukturálně popsaných komponent – více ve vlastní práci.

Pomocí vývojového prostředí Xilinx ISE 10.1 byly syntetizovány zdrojové kódy behaviorálně a strukturálně popsaného kodeku. Mezi nejvýznamnější kritéria pro hodnocení a porovnávání číslicových systému patří jejich rychlost a plocha. Byla použita struktura FPGA s čipem Spartan-3 do níž je a aplikován Iwadari-Masseyho kodek. Z Obr. 6.4 je zřejmé, že plocha na čipu spotřebovaná logikou tvořící kodér či dekodér Iwadari-Masseyho kódu se zvyšující se hodnotou korekční schopností prudce roste, což je v souladu s teoreticky vyjádřenými vztahy pro počet paměťových buněk a logických operátorů. Objem spotřebované logiky není příliš závislý na tom, zda je komponentou kodér, či dekodér a zda byla syntetizována z behaviorálního či strukturálního popisu. V Obr. 6.3 je zaznamenána závislost maximální dosažitelné pracovní frekvence komponent v závislosti na hodnotě generického parametru. Je zřejmé, že hodnoty pracovní frekvence klesají se zvyšující se korekční schopností a že strukturálně popsané komponenty v tomto ohledu poskytují výrazně lepší vlastnosti.



Obr. 6.3: Závislost maximální pracovní frekvence kodeku.



Obr. 6.4: Závislost objemu použité logiky kodeku.

Rozsah i způsob popisů cílových komponent v behaviorálním a strukturálním stylu se velice liší přestože, vycházely oba popisy ze stejných předloh viz Obr 4.3 a Obr. 4.4. Avšak ekvivalence činnosti komponent byla ověřena a potvrzena v předchozích částech. I charakteristické parametry komponent popsané v těchto dvou stylech jsou velmi blízké. Definitivní posouzení odlišností poskytuje porovnání RTL schémat vygenerovaných při syntéze. Po podrobném prozkoumání schémat je zřejmé, že jsou si výsledné obvody velmi podobné, v jistých částech jsou často naprosto identické. Pravděpodobně jediným místem, kde se výsledná schémata popsaná v různých stylech liší, jsou obvody realizující vlastní korekci bitů. Ze schématu pro behaviorálně popsaný dekodér je jen velmi těžko rozpoznatelné, které obvody se přesně na této činnosti podílejí. Oproti dobře čitelnému a logicky strukturovanému schématu pro strukturálně popsaný dekodér.

7 ZÁVĚR

Zabezpečovací kódování patří k nepostradatelné části téměř každého přenosového systému. Jelikož při přenosu informace kanálem dochází k jejímu poškození kvůli různorodým vlivům šumu. Problémem shluků chyb se v současných systémech musí zabývat stále více zařízení. Hlavní příčinu představuje rychlý rozmach vysokorychlostních systémů pro výměnu, zpracování i uchování dat. K potlačení shluků chyb či k opravě chybných bitů se využívají rozličné metody, jejichž stručná charakteristika i principy byly představeny v úvodu.

Pro rozšíření vhodných použitelných systémů pro uplatnění v oblasti korekce shlukových chyb je využito alternativního řešení korekce shlukových chyb pomocí

vybraných systematických konvolučních kódů. Jelikož v individuálních protichybových systémech je vhodné zvážit použití jiných zabezpečovacích kódových technik než masová aplikace stávajících robustních řešení pro dosažení maximální efektivity, které lze dosáhnout v některých případech i s jednoduššími systémy. Proto pro soubor systematických konvolučních kódů je proveden rozbor i srovnání jejich zabezpečovacích schopností. Účelem je poskytnout adekvátní náhradu hromadně používaných systémů pro opravu shluků chyb, jejichž nasazení v individuálních protichybových systémech umožní zlepšení přenosu dat. Odvozen je podrobný matematický aparát pro rozšíření souboru kritérií, která umožní vhodnější srovnání protichybových systémů pro přenos digitálních signálů. Získané vztahy s využitím blokové vytvářecí matice mají všeobecnou platnost. Uplatnění naleznou při návrzích individuálních protichybových systémů, kdy bez předchozí podrobné znalosti daného systému se získá více popisných informací.

Na základě konfrontace s jinými nejčastěji aplikovanými způsoby ochrany zprávy před shlukem chyb se potvrdila existence mezery používaných systémů pro korekci shluků chyb různé délky mezi systémy používající základní blokové kódy na straně jedné a sofistikovanými systémy na straně druhé. Dle získaných výsledků lze v individuálních protichybových systémech dosáhnout lepších vlastností využitím skupiny konvolučních kódů. Kvůli prověření jejich správnosti jsou teoretické výsledky podrobeny kontrole pomocí simulace v grafickém uživatelském rozhraní Matlab. Účelem je též dostatečně názorné přiblížení jednotlivých fází kódovacích i dekódovacích postupů. Obdržené výsledky simulace potvrzují správnou funkci příslušného kodeku. Tedy v kapitole 4.2, 4.3, 4.4 odvozená rozšiřující kritéria popisu kódů byla realizovanou simulací dostatečně prověřena. Veškeré takto získané výsledky jsou hojně publikovány.

Na závěr je ověřena možnost realizace individuálních protichybových systémů, kdy je třeba uvažovat součinnost zařízení nejen pro bezprostřední realizaci kódu. I při výběru vhodné metody pro hardwarovou implementaci záleží na vlastnostech, na které se klade hlavní důraz, jelikož mnohé požadavky vystupují proti sobě a je třeba hledat kompromis. Vzhledem k další využitelnosti získaných výsledků, zefektivňování procesu návrhu i realizaci číslicových systémů a jejich snadné přenositelnosti je možný způsob realizace demonstrován pomocí programově logických obvodů. Analýza struktury pomocí popisu VHDL představuje vlastní systémy realizace protichybového kódování. Takto vytvořený systém je již možné použít a zařadit do nadřazených přenosových struktur. Opět se potvrdily teoreticky získané výsledky a též se potvrdila ekvivalence činnosti komponent v behaviorálním i strukturálním stylu, přestože jejich popis se značně liší. Navržený rozšířený matematický popis systematických konvolučních kódů i možnost jejich realizace v individuálních protichybových systémech lze pokládat za dostatečně prověřený. Všechny nově získané poznatky z této práce, lze uplatnit u návrhů individuálních protichybových systémů zaměřených na korekci shluků chyb.

8 LITERATURA

[1] BASALAMAH, A., SATO, T. A Comparison of Packet-Level and Byte-Level Reliable FEC Multicast Protocols for WLANs. *In Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, 2007, pp. 4702 - 4707, ISBN 978-1-4244-1043-9.

[2] BLAUM, M. Enhanced decoding of interleaved error correcting codes. *IEEE International Symposium on Information Theory*, 1995, pp. 412 – 414, ISBN 0-7803-2453-6.

[3] CELANDRONI, N. Comparison of FEC types with regard to the efficiency of TCP connections over AWGN satellite channels. *In Wireless Communications, IEEE Transactions*, 2006, pp. 1735 - 1745, ISSN 1536-1276.

[4] CLARK, C., CAIN, B. *Error-Correction Coding for digital communications*. New York: Plenum Pr., 1982, 436 p., ISBN 0306406152.

[5] EROZAN, K., BANE, V. *Advanced error control techniques for data storage systems*. Boca Raton, FL: CRC Taylor & Francis, 2006, 288 p., ISBN: 0-8493-9547-X.

[6] FARRUGIA, R., A., DEBONO, C., J. A Statistical Bit Error Generator for Emulation of Complex Forward Error Correction Schemes. *In Communications, 2007. ICC '07. IEEE International Conference*, 2007, Glasgow, pp. 177 - 182, ISBN 1-4244-0353-7.

[7] FENG, W., YUAN, J. A code-matched interleaver design for turbo codes. *Communications, IEEE Transactions*, 2002, Vol. 50, Issue 6, pp. 926 – 937, ISSN 0090-6778.

[8] GOFF, S., KHOO, K., TSIMENIDIS, C. Constellation Shaping for Bandwidth-Efficient Turbo-Coded Modulation With Iterative Receiver. *IEEE Transactions on Wireless Communications*, 2007, Vol. 6, pp. 2223 – 2233, ISSN 1558-2248.

[9] HEEGARD, CH., WICKER, S. *Turbo Coding*. 3 edition, Springer; 2000, 240 p., ISBN-0-7923-8378-8.

[10] JAIKWAN, J. *The Simulation, Modeling and Analysis of Wireless Local Area Networks Supporting the IEEE 802.11 Standard*. Storming Media, 1998, 101 p., ISBN 1423555031.

[11] JEONGSEOK, H., MCLAUGHLIN, W. Low-density parity-check codes over Gaussian channels with erasures. *Information Theory, IEEE Transactions*, 2003, Vol 49, Issue 7, pp. 1801 – 1809, ISSN 0018-9448.

[12] JURCA, D., FROSSARD, P. Optimal FEC rate for media streaming in active network. *In Multimedia and Expo, 2004. ICME '04. IEEE International Conference*, 2004, Taipei, Vol. 2, pp. 1319 - 1322, ISBN 0-7803-8603-5.

[13] LIU, H. et al. Staggered FEC System for Seamless Handoff in Wireless LANs: Implementation Experience and Experimental Study. *In Multimedia, ISM 2007. Ninth IEEE International Symposium*, 2007, Taichung, pp. 283 - 290, ISBN 978-0-7695-3058-1.

[14] MUKARAMI, K. *VoIP Evaluation for MBWA*. Project IEEE 802.20 Working Group on Mobile Broadband Wireless Access, 2005, 11 p.

[15] NEEB, C., THUL, M., WEHN, N. Network-on-chip-centric approach to interleaving in high throughput channel decoders. *IEEE International Symposium Circuits and Systems*, 2005, pp. 1766 – 1769, ISBN 0-7803-8834-8.

[16] NĚMEC, K. *Kódové zabezpečovací systémy*. Teze habilitační práce, ÚTKO FEI VUT Brno, Brno 1999, 34 s., ISBN 80-214-1312-3.

[17] SHIINIZU, K., et al. Reconfigurable adaptive FEC system with interleaving. *In Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific*, 2005, Vol. 2, pp. 1252 - 1255, ISBN 0-7803-8736-8.

[18] STOERTE, C., MACHMERTH, M. Implementation of error decoders for A-VSB systems with additional use of transport stream information as forward error correction. *In Devices, Circuits and Systems, 2008. ICCDCS 2008. 7th International Caribbean Conference*, 2008, Cancun, pp. 1 - 4, ISBN 978-1-4244-1956-2.

[19] THUL, M., GILBERT, F., VOGT, T. A Scalable System Architecture for High-Throughput Turbo-Decoders. *Journal of VLSI Signal Processing Systems*, 2005, Vol. 39, Issue 1 - 2, pp. 63 – 77, ISSN 0922-5773.
[20] THUL, M., WEHN, N. FPGA implementation of parallel turbo-decoders. *17th Symposium on Circuits and Systems Design*, 2004, pp. 198 – 203, ISBN 1-58113-947-0.
[21] VLČEK, K. *Komprese a kódová zabezpečení v multimediálních komunikacích.* 2. vyd., BEN, Praha 2004, 258 s., ISBN 80-7300-134-9.
[22] WILSON, G. *Digital Modulation and Coding, Englewood Cliffs*, Prentice - Hall, 1996, 712 p., ISBN 0-13-210071-1.

Vlastní publikační činnost související s disertační prací:

[23] KŘIVÁNEK, V., ČÍKA, P. Simulace shlukových chyb v Matlabu. *Elektrorevue - Internetový časopis* (http://www.elektrorevue.cz), 2006, roč. 2006, č. 35, s. 1 - 10, ISSN 1213-1539.
[24] KŘIVÁNEK, V., ČÍKA, P. Výběr nejvhodnějšího konvolučního kódu - I. *Access Server*, 2007, roč. 5, č. 3, s. 1 - 8, ISSN 1214-9675.
[25] KŘIVÁNEK, V., KYSELÁK, M. Výběr nejvhodnějšího konvolučního kódu - II. *Access Server*, 2007, roč. 5, č. 3, s. 1 - 6, ISSN 1214-9675.
[26] KŘIVÁNEK, V. Oprava shlukových chyb - Iwadariho kód. In *Elektrotechnika a informatika 2005*. Plzeň: Západočeská univerzita v Plzni, 2005. s. 45 - 48, ISBN 80-7043-374-4.
[27] KŘIVÁNEK, V. The Use of Matlab for the Simulation of the Burst Error Correction. *International Journal of Computer Science and Network Security*, 2006, Vol. 6, No. 7B, pp. 141 - 145, ISSN 1738-7906.
[28] KŘIVÁNEK, V. Návrh kodérů a dekodérů umožňující opravu shlukových chyb a jejich simulace. *Elektrorevue - Internetový časopis* (http://www.elektrorevue.cz), 2006, roč. 2006, č. 8, s. 1 - 9, ISSN 1213-1539.
[29] KŘIVÁNEK, V. Verification of the error-control security process by means of simulation. *International Transactions on Communication and Signal Processing*, 2006, Vol. 9, No. 1, pp. 128 - 136, ISSN 1738-9682.
[30] KŘIVÁNEK, V. Využití počítačové simulace pro výuku principů zabezpečovacích kódů. In *5. ročník konference Alternativní metody výuky*. GAUDEAMUS, Universita Hradec Králové: Univerzita Karlova v Praze, 2007, s. 1 - 4, ISBN 978-80-7041-129-2.
[31] KŘIVÁNEK, V. Computer Simulation Forward-Eror-Corection Principles. In *8th International Conference, Research in Telecommunication Technology RTT - 2007 Liptovský Ján*, *SR*. 2007, pp. 209 - 214, ISBN 978-80-8070-735-4.
[32] KŘIVÁNEK, V. Ochrana dat před shluky chyb, Berlekamp-Preparatův kód. *Elektrorevue - Internetový časopis* (http://www.elektrorevue.cz), 2007, roč. 9, č. 49, s. 1 - 7, ISSN 1213-1539.
[33] KŘIVÁNEK, V. Correction Error Data Rising in the Transmission Channel. *International Transaction on Computer Science and Engineering*, 2007, Vol. 43, No. 1, pp. 9 - 16, ISSN 1738-6438.
[34] KŘIVÁNEK, V. Simulace korekčních kódů - QoS. In *6. ročník konference Alternativní metody výuky*. Přírodovědecká fakulta UK, Praha: Univerzita Karlova v Praze, 2008, s. 1 - 6, ISBN 978-80-7041-454-5.
[35] KŘIVÁNEK, V. Enhancement in the Protection of Transmitted Data. *International Journal of Computer Science and Network Security*, 2008, Vol. 8, No. 7, pp. 95 – 98, ISSN 1738-7906.
[36] KŘIVÁNEK, V. Comparison of the convolutional codes design complexity. *International Transaction on Computer Science and Engineering*, 2008, Vol. 47, No. 1, pp. 43 - 48, ISSN 1738-6438.

CURRICULUM VITAE

Osobní údaje

Jméno:	Vítězslav Křivánek
Datum a místo narození:	1. června 1981 v Třebíči
Email:	krivan@feec.vutbr.cz, V.Krivanek@seznam.cz

Vzdělání

od 2005	**VUT v Brně** Fakulta elektrotechniky a komunikačních technologií Doktorské studium Obor: Teleinformatika Téma disertační práce: Systémy realizace protichybového kódování.
2000-2005	**VUT v Brně** Fakulta elektrotechniky a komunikačních technologií Magisterské studium Obor: Elektronika a sdělovací technika
1993-2000	Gymnázium Vídeňská, Brno (maturita)

Další vzdělání

2007	Osvědčení Hlavní vedoucí dětských táborů
2006	Vedoucí oddělení vědy a techniky pro každého na Junior DDM Brno – pedagogický pracovník
2005	Osvědčení Kurz vedoucích herních klubů
2004	Invention Machine - TRIZ Certificate
2003	Autodesk Academia Certificate - AutoCAD 2002
2002	Osvědčení Zdravotník zotavovacích akcí
2000	Osvědčení Vedoucí dětského kolektivu

Ostatní

- Účast na projektech:

  GAČR reg. č. GA102/03/0762 „Analýza přenosových parametrů xDSL systémů pomocí reálných přístupových sítí“ (spoluřešitel).

  NBÚ reg. č. ST200520005002 „Synchronizace blokových šifer pro modulární kryptografický systém pro verzi BRI ISDN a PRI ISDN“ (spoluřešitel).

  FRVŠ reg. č. 256/2006/F1a „Inovace způsobu uskutečňování přednášek předmětu Datová komunikace“ (spoluřešitel).

  MPO ČR reg. č. FT-TA3/001 „Výzkum a vývoj obousměrné komunikační technologie pro varování obyvatelstva“ (spoluřešitel).

  FRVŠ reg. č. 993/2007/G1 „Simulace zabezpečovacích korekčních kódů a její začlenění do výuky předmětu Datová komunikace“ (spoluřešitel).

  FRVŠ reg. č. 1663/2007/F1a „Podpora výuky předmětu zaměřeného na programování v jazyce C++ pomocí interaktivních kontrolních testů“ (spoluřešitel).

  FRVŠ reg. č. 1388/2008/F1a „Zdokonalení výuky zabezpečování přenosu informace“ (spoluřešitel).

  FRVŠ reg. č. 1775/2008/G1 „Zvýraznění problematiky kvality služeb v předmětu Služby telekomunikačních sítí“ (spoluřešitel).

  NPV II. reg. č. 2C08002 „KAAPS „Výzkum univerzální a komplexní autentizace a autorizace pro pevné a mobilní počítačové sítě“ (spoluřešitel).
- Autor, resp. spoluautor 22 publikací vztahujících se k řešené problematice.

ABSTRACT

Due to growing transmission speed burst-forming errors tend to occur still more frequently not exclusively in data transmission. The presented paper concentrates on the search for alternative burst error correction solutions complementing the existing methods in use. Its objective is an elaboration of a detailed analysis of the issue of convolution codes for error burst correction which can be used in individual anti-error systems and thus an achievement of better results than those attained by mass application of the existing solutions.

First the methods implemented to remove or suppress burst errors are briefly characterized. This part is followed by a detailed description of the individual systematic convolution codes by means of mathematical tools which extend the set of possible evaluative criteria of anti-error systems which can be applied while assessing proposals for individual solutions. The acquired code properties are compared with convolution codes as well as with other versions of proposals for message protection against an error burst. The processed convolution codes are subject to testing by means of Matlab mathematical programme simulation in order to validate the correctness of the derived mathematical tools. This is because simulation represents the principal method applied to verify and present an already proposed security process and enables the acquisition of a better overview of the issue at hand. The feasibility of the individual anti-error systems is then confirmed by way of creating a circuit behaviour description in the VHDL language. Its high portability presents a big advantage when drafting individual systems of the actual implementation.